Journal of Computer-Aided Molecular Design
https://doi.org/10.1007/s10822-019-00274-0

Validating the validation: reanalyzing a large-scale comparison of deep learning and machine learning models for bioactivity prediction

Matthew C. Robinson[1] · **Robert C. Glen**[2,3] · **Alpha A. Lee**[1]

Received: 27 May 2019 / Accepted: 22 December 2019

Abstract
Machine learning methods may have the potential to significantly accelerate drug discovery. However, the increasing rate of new methodological approaches being published in the literature raises the fundamental question of how models should be benchmarked and validated. We reanalyze the data generated by a recently published large-scale comparison of machine learning models for bioactivity prediction and arrive at a somewhat different conclusion. We show that the performance of support vector machines is competitive with that of deep learning methods. Additionally, using a series of numerical experiments, we question the relevance of area under the receiver operating characteristic curve as a metric in virtual screening. We further suggest that area under the precision–recall curve should be used in conjunction with the receiver operating characteristic curve. Our numerical experiments also highlight challenges in estimating the uncertainty in model performance via scaffold-split nested cross validation.

Introduction

Computational approaches to drug discovery are often justified as necessary due to the prohibitive time and cost of experiments. Unfortunately, many papers fail to sufficiently prove that the proposed, novel techniques are actually an advance on current approaches when applied to realistic drug discovery programs. Models are often shown to work in situations differing greatly from reality, producing impressive metrics that differ greatly from the quantity of interest. It is then often the time and cost of properly implementing and testing these proposed techniques against existing methods that becomes prohibitive for the practitioner. There is also the significant opportunity cost if models prove to be inaccurate and misdirect resources.

These concerns are not new to the field of computational chemistry. Walters [1], Landrum and Stiefel [2], and others have previously critiqued the state of the literature, even referring to many papers as "advertisements". Furthermore, Nicholls has provided useful overviews of statistical techniques for uncertainty quantification and method comparison [3–5]. Recent works provided an important review on the importance of evaluating bias in data and the implications for deep learning algorithms in virtual screening [6, 7]. However, many authors still neglect the relevant issues, and results are frequently reported without error bars, proper train and test set splitting, and easily usable code or data.

Problems with validation are not unique to chemoinformatics or computational chemistry: numerous papers and manuscripts in the machine learning literature have been devoted to the proper evaluation of methods, with special concern to the applicability of statistical testing procedures for method comparison [8–12]. Recent reviews also provide background into procedures for hyperparameter optimization and model selection [13]. However, despite this work, researchers still find that new approaches frequently exploit

Electronic supplementary material The online version of this article (doi:https://doi.org/10.1007/s10822-019-00274-0) contains supplementary material, which is available to authorized users.

✉ Alpha A. Lee
aal44@cam.ac.uk

[1] Department of Physics, J J Thomson Avenue, Cambridge CB3 0HE, UK

[2] The Centre for Molecular Informatics, Department of Chemistry, University of Cambridge, Cambridge CB21EW, UK

[3] Computational and Systems Medicine, Department of Metabolism, Digestion and Reproduction, Faculty of Medicine, Imperial College, South Kensington, London SW72AZ, UK

Published online: 20 January 2020

bias in the training set, likely overfit to benchmark datasets [14], and even find that thousands of papers may be based on an initial result that was simply statistical noise [15].

These errors are rampant for a number of reasons. One overarching issue is that the literature has many differing suggestions and involves theoretical statistical ideas—different metrics reward different aspects of the model, and commonly used metrics in machine learning do not necessarily reward models that are useful in drug design. Moreover, even when practitioners are interested in a thorough analysis of results, the task can be quite time intensive and costly. To properly test a new neural network architecture against older methods using several different random seeds, dataset splits, and learning rates might take on the order of 1000s of GPU hours and become a monetary concern.

Therefore, we find that there is a clear need for research regarding the proper way to compare models in computational chemistry. That is, we need to first ask the meta-question of how to ask: which model is the best? For many research areas, including bioactivity prediction, past work has revolved around proving state-of-the-art (SOTA) performance on a set of benchmark datasets [16]. While benchmark datasets are important and efforts such as ImageNet have been revolutionary for the computer vision and NLP communities, their use has not been without controversy [17]. When SOTA performance on these benchmarks becomes the main goal of algorithm development, there becomes less focus on understanding the robustness and domains of applicability for each model, while interesting, potentially fruitful ideas failing to achieve SOTA may be ignored. As a result, members of the NLP and computer vision communities have called for a renewed focus on careful dataset creation, such as constructing difficult "adversarial datasets" probing how certain models fail [18]. If a field is to indeed prioritize the optimization of model performance on a small selection of datasets, care should be taken to ensure that these datasets and performance metrics are proper surrogates for the more general problem of interest.

In this paper, we examine this question by reanalyzing the recent validation study by Mayr and coworkers [19]. Our study is made possible by their extensive effort in building a large-scale benchmarking study, as well as their generosity in making the code and data publicly available. The questions we will ask are: (1) Is one machine learning method significantly better than the rest, using metrics adopted by Mayr et al.? (2) Are the metrics adopted by Mayr et al. the most relevant to ligand-based bioactivity prediction? Our key conclusion is an alternative interpretation of their results that considers both statistical *and* practical significance—we argue that deep learning methods do not significantly outperform all competing methods. We also show, via a series of examples, that the precision–recall curve is relevant to ligand-based drug discovery and should be used in combination with the ROC–AUC metric. In reaching these conclusions, we also discuss and review issues of uncertainty and model comparison that are central to the field.

The source code used for our reanalysis is available on GitHub https://github.com/mc-robinson/validating_validation_supp_info

Study design of Mayr et al.

Our study is motivated by the recent paper, entitled "Large-scale comparison of machine learning methods for drug target prediction on ChEMBL", by Mayr and coworkers. Realizing the recent success of deep learning in other fields and its introduction into drug-discovery [20, 21], Mayr and coworkers performed a large-scale evaluation of the method's success against other commonly-used machine learning methods in the drug discovery community. Their goal was to combat three common problems with model evaluation in chemical prediction: (1) a lack of large scale studies, (2) compound series bias in testing of drug-discovery algorithms, and (3) bias in hyperparameter selection.

The Mayr et al. evaluation, based entirely on ligand-based approaches, had the explicit goal of comparing "the performance of deep learning with that of other methods for drug target prediction." In pursuing this goal, the authors cited the relatively small number of assays in previous evaluation studies such as MoleculeNet [21] and the need for larger scale evaluation. The authors believe that these small studies "restrict the conclusions of the method comparisons to a certain subset of assays and underestimate the multitask learning effect in spite of the large amount of data being available publicly." To correct this shortcoming, Mayr et al. extract data including roughly 456,000 compounds and over 1300 assays from ChEMBL [22].

Notably, the ChEMBL data is quite heterogeneous. The diverse set of target classes includes ion channels, receptors, transporters, transcription factors, while the similarly diverse assay types include ADME, binding, functional, physiochemical, and toxicity assays. The number of compounds in each assay is also quite variable—ranging from roughly 100 compounds to over 30,000 in a given assay. In order to treat each problem as a separate binary classification procedure, the authors also develop a procedure to automatically convert the assay measurements to binary labels. Each assay is then treated as an individual classification problem.

The compounds were then featurized using several different schemes including toxicophore features, semisparse features, depth first search features, and the popular (ECFP6) fingerprint [16]. In our study, we chose to examine the ECFP6 fingerprint, as implemented by Mayr et al. in jCompoundMapper, because the similar Morgan fingerprint with

radius 3 can easily be constructed from molecular data using the popular open-source program RDKit [23].

Is there a best model?

In their study, Mayr et al. concluded that "deep learning methods significantly outperform all competing methods." Much of this conclusion is based upon small *p*-values resulting from a Wilcoxon signed-rank test used to quantify the differences in the average performance of the classifiers. For example, they report a *p*-value of 1.985×10^{-7} for the alternate hypothesis that feedforward deep neural networks (FNN) have a higher area under the receiver operating curve (AUC–ROC) than support vector machines (SVM). For the alternative hypothesis that FNN outperform Random Forests (RF), the p-value is even more extreme (8.491×10^{-88}). From such low *p*-values, one might be led to believe that FNN is the only algorithm worth trying in drug discovery. Yet, a closer look at the data reveals that this conclusion is clearly erroneous and obscures much of the variability from assay to assay.

Are all assays created equal?

To demonstrate the problems, we begin with an initial example of SVM and FNN performances using ECFP6 fingerprints. Table 1 shows AUC–ROC results from the FNN and SVM classifiers for two assays in the Mayr et al. dataset. Assay A is a functional assay consisting of a small number of samples. Each fold is heavily imbalanced and consists mostly of active compounds. Importantly, this is often the opposite imbalance one would observe in a real screen. As is expected with a small amount of highly imbalanced data, both the FNN and SVM classifier show highly variable results with very large confidence intervals. In fold 2, where only a single active compound is present in the test set, it is not even clear how to calculate the confidence intervals for AUC–ROC. The mean and standard error of the mean (SEM) are also calculated over the threefold, though this is slightly dangerous since it discards all of our knowledge of uncertainty in each fold.

In contrast, assay B is a functional assay with large samples and imbalances that more closely resemble those typically seen in the literature. Performance is quite good, with the SVM classifier outperforming the FNN classifier on each fold. Furthermore, the confidence intervals for each AUC–ROC value are quite small. Again the mean and SEM are calculated across the folds for each classifier. Additionally, Fig. 1 gives a visual representation of the performances for assay A and assay B.

One would likely agree that Fig. 1 shows a striking difference between the results of the two assays. While the results of assay A for FNN and SVM are extremely noisy and raise many questions, assay B shows a well-defined difference in the performance of the two algorithms, even relative to the noise levels of the measurements. Though not a formal analysis, due to the presence of noise, one would likely consider the difference in mean performances on assay A, $0.67 - 0.57 = 0.10$, to be much less meaningful than the difference in mean performances on assay B, $0.929 - 0.900 = 0.029$. To most practitioners, comparative performances on assay B would give much more evidence to SVM outperforming FNN than the comparative performances on assay A, even though it is lower in magnitude. More formally, one can compare the effect size, which measures the difference in the performance of the two models relative to the standard deviation of the data, and thus accounts for the variability of the respective datasets. The effect size can also be related to the probability of one model outperforming the other [24]. Note that the *p*-value

Table 1 Two separate assays from the Mayr et al. data with the accompanying FNN and SVM prediction results

	Fold 1	Fold 2	Fold 3	MEAN	SEM
A: ChEMBL 1964055					
FNN AUC–ROC (95% CI)	0.44 (0.035, 0.94)	0.62 (0.0, 1.0)*	0.64 (0.34, 0.86)	0.57	0.05
SVM AUC–ROC (95% CI)	0.38 (0.02, 0.94)	0.97 (0.0, 1.0)*	0.68 (0.38, 0.88)	0.67	0.14
Test set size (actives/ inactives)	35 (32/3)	30 (29/1)	35 (29/6)		
B: ChEMBL 1794580					
FNN AUC–ROC (95% CI)	0.889 (0.883, 0.895)	0.905 (0.900, 0.910)	0.906 (0.900, 0.911)	0.900	0.005
SVM AUC–ROC (95% CI)	0.926 (0.921, 0.931)	0.926 (0.921, 0.930)	0.934 (0.930, 0.939)	0.929	0.002
Test set size (actives/ inactives)	19388 (5553/13855)	25165 (6918/18247)	19363 (5491/13872)		

Confidence intervals for AUC–ROC are calculated through the Hanley–Mcneil method while the standard error of the mean (SEM) across folds is calculated in the standard fashion

*Indicates that the confidence interval is calculated using a different, simulation based approach because it is not possible to calculate the effective degrees of freedom in the usual way, when only one sample is given from the positive class

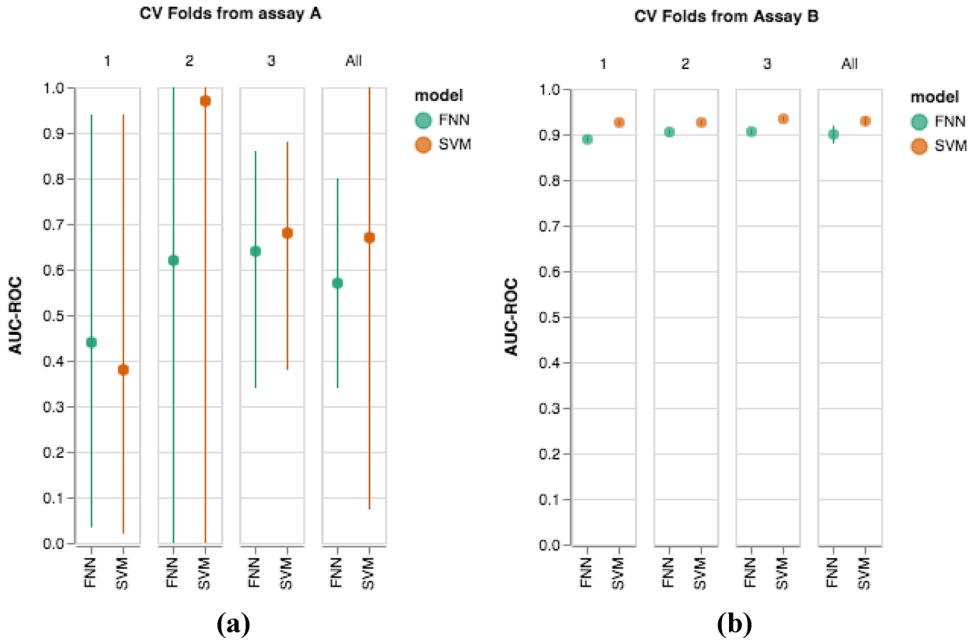

Fig. 1 Figures displaying the information contained in Table 1. In the case of fold 2 for assay A, 95% confidence intervals are calculated based on a simulation approach. For the mean values across all folds, a *t*-distribution with two degrees of freedom is used to calclulate 95% confidence intervals for the mean AUC–ROC across the threefold

is dependent on sample size whereas the effect size is not—with a sufficiently large sample, relying on *p*-values will almost always suggest a significant difference, unless there is no effect whatsoever (effect size is exactly zero). However, very small differences, even if significant, are practically meaningless. Using the canonical definition for effect size from Cohen (Cohen's d):

$$d = \frac{\mu_2 - \mu_1}{\sqrt{\frac{s_1^2 + s_2^2}{2}}},$$

where μ_1 and μ_2 are the mean performances and s_1 and s_2 are the standard deviations of the data, not the means. Using this formula, the effect size in assay B (4.40) is approximately eight times the size of the effect size in assay A (0.55). Nevertheless, a problem arises because the Wilcoxon signed-rank test used by Mayr treats the noisy, less informative assay A as greater evidence than assay B for the superiority of SVMs over FNNs.

The Wilcoxon signed-rank test is a non-parametric paired difference test often used when determining if matched samples came from the same distribution. The test is perhaps best explained by example: imagine two methods, M_{shallow} and M_{deep} are used on a variety of prediction tasks. These prediction tasks vary widely and the algorithms have vastly different expected performances on each task. For our example, consider the sample tasks of predicting a coin flip (COIN), predicting the species of a flower (FLOWER), and predicting the label of an image (IMAGE) among many other varying tasks. The tests to verify the accuracy (ACC) on these differing tasks are also quite different. For the coin prediction tasks, only ten coins are tossed. For the flower task, around 100 flowers are used. And for the image task

Table 2 The results form our imagined example

	ACC_{shallow}	ACC_{deep}	$ACC_{\text{shallow}} - ACC_{\text{deep}}$		
			Absolute difference	Sign	Signed Rank
...	...	...	...	...	...
COIN	0.6 (0.45, 0.75)	0.4 (0.25, 0.55)	0.2	+	+ 20
...	...	...	...	...	...
FLOWER	0.98 (0.97, 0.99)	0.99 (0.97, 1.0)	0.01	−	− 2
...	...	...	...	...	...
IMAGE	0.894 (0.891, 0.897)	0.941 (0.938, 0.944)	0.057	−	− 7
...	...	...	...	...	...

The signed-rank is simply the rank of the difference in $ACC_{\text{shallow}} - ACC_{\text{deep}}$ among all such differences multiplied by the sign of the difference

10,000 images are used. Thus, our made up results when comparing the models may look like so (Table 2):

As our imagined table shows, the difference in performance between the methods is most drastic for the COIN dataset, which is also the most noisy, as shown by the large confidence intervals. However, this result is also the most meaningless of the three shown, since we know that all methods will eventually converge to an accuracy around 0.5 in the large number of test samples. The IMAGE dataset is likely the best indicator of the superior method (at least on image problems), but the "rank" of the difference (after ordering all the absolute differences) is quite small compared that of the COIN, and potentially many other unreliable tests of performance. Unfortunately, since the Wilcoxon signed-rank test only relies on the signed-rank (rank of difference multiplied by the sign of the difference), all information regarding the variability in a given test is discarded.

The null hypothesis of the Wilcoxon test is that the differences in the methods are distributed symmetrically and centered on zero. The test statistic W is simply the sum of the signed ranks and has an expected value of zero, with a known variance. As a result, the larger magnitude differences between the two methods will be considered more important by the test, due to their high ranks. Unfortunately, in our illustrative example, the highly ranked differences are not those that give the best evidence of differences between the methods.

Coming back to the example from Mayr and coworkers, the test treats the difference in performances on each assay as commensurate, and assumes that the larger magnitude difference of mean AUC–ROC values in assay A should carry more weight than the smaller magnitude difference of mean AUC–ROC values in assay B. This, again, is not necessarily true.

Instead, effect size, which measures the magnitude of the difference relative to the uncertainty, is more important than pure magnitude differences. As another complication, differences in AUC/probability space are not straightforward: $p = 0.01$ and $p = 1 \times 10^{-6}$ have a smaller difference in absolute magnitude than $p = 0.51$ and $p = 0.52$; however, the difference between one in 100 and one in a million is likely much more important than the difference between 51 and 52%. Lastly, these concerns aside, assuming commensurate results were already problematic given the heterogeneity of the assay types, changing sample sizes, varying imbalances, and diverse target classes.

A different test, a different question

Having realized that the Wilcoxon signed-rank test is inappropriate, we turn to the sign test as perhaps the most appropriate procedure. The sign test essentially counts the proportion of "wins" for a given algorithm over another on all of the datasets. That is, we simply consider the sign of the difference in performance between the methods. The test allows us to probe the question: "on a given dataset/assay, which of the methods will perform the best?" This question addresses the concern of a practitioner implementing a bioactivity model and considering many potential predictive models. In contrast, the statistic of the Wilcoxon signed-rank test is much less interpretable, providing less clarity to the user.

As with many of these tests, the null hypothesis of the sign test is that the two models show the same AUC–ROC performance on the datasets. Assuming this null hypothesis, the algorithm displaying the better performance on a given assay should be determined by a coin-flip. Therefore, given N assays, we expect each classifier to win on approximately $N/2$ assays. In our illustrative example, if both M_{shallow} and M_{deep} were tested on 100 different datasets, each method would be expected to "win", i.e. outperform the other method, on approximately 50 of the datasets. Obviously, variability is expected, which can be quantified by deviation from the expected binomial distribution.

There are, of course, still problems with the sign test. First, the test still discards most of the uncertainty information. Secondly, the test still counts the assays in Table 1 and Fig. 1 of equal weight, which is better than in a rank test, but still suboptimal. Additionally, due to the lack of parametric assumptions, the sign test has low power, meaning that it often fails to detect a statistically significant difference in algorithms when one exists.

Using the sign-test we calculated 95% Wilson score intervals for the sign-test statistic for the alternative hypothesis that FNN has better AUC–ROC performance than SVM, the second best performing classifier according to Mayr et al. Using all 3930 test folds in the analysis (since each is indeed an independent test set) gives an interval of (0.502, 0.534), while only comparing the mean AUC values per assay gives a confidence interval of (0.501, 0.564). While both of these tests are narrowly significant at the $\alpha = 0.05$ level (intervals do not include 0.5), it is worth examining the practical meaning of these results.

According to the statistic, our data is compatible with an FNN classifier beating an SVM classifier on 50% to 56% of the assays, Thus, if one were to conclude that only an FNN classifier is worth trying, the user would be failing to use a better classifier almost 50% of the time! And this is in the case of a two classifier comparison. Considering all the classifiers, FNN and SVM both perform the best in 24% of the assays, while every other classifier considered by Mayr et al. is the best performing classifier on at least 5% of the assays (Table 3 shows a breakdown of wins). Clearly, some of these results are just noise due to small assay sizes; however, it indicates that classifier performance is likely assay dependent, and one should try multiple classifiers for a given

Table 3 The percent of test folds, across all assays, that a method is the best performing method

Method	SVM	FNN	GC	NB	LSTM	RF	Weave	KNN	Tie
% of folds that the method is the best performing method	24.6	24.6	9.99	9.69	7.91	7.71	7.51	5.73	2.34

SVM and FNN are clearly the best performing methods, and it is noteworthy that SVM outperforms deep learning methods such as GC, LSTM and Weave

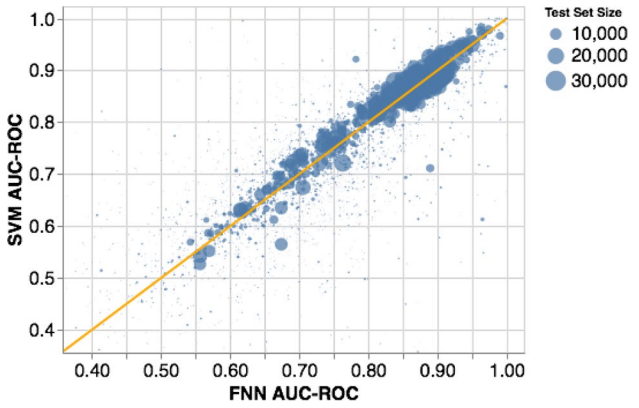

Fig. 2 A comparison of FNN and SVM AUC–ROC performance on all test folds. The orange line indicates the identity line of slope 1, while the dot size indicates the size of the test set

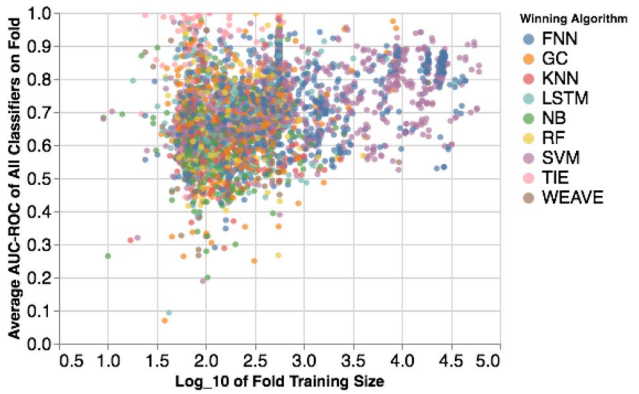

Fig. 3 The best performing algorithm (indicated by color) in terms of AUC–ROC for all test folds as the training size increases

problem. (It is also noteworthy that the dataset comprises many assay types, e.g. enzyme inhibition/binding affinity/efficacy, which are qualitatively different.)

Unfortunately, though the sign test may be an improvement, the act of averaging results over many heterogeneous assays still fails to properly quantify the applicability and robustness of each model. Reporting merely the average performance occludes the success of each method on assays with differing makeups of actives and inactivates, similarities among molecules, and levels of noise.

The above considerations are illustrated by the data in Figs. 2 and 3. Figure 2 shows that the FNN and SVM performance is almost identical for large datasets, but the difference between the performances varies quite sporadically for assays with fewer compounds (the smaller points in the figure). Additionally, Fig. 3 shows the best performing algorithm for each independent test fold; we also plot the other algorithms that Mayr et al. considered, namely random forest (RF), k-nearest neighbours (kNN), Graph Convolutional neural networks (GC), Weave, and Long Short-Term Memory networks with SMILES input (LSTM). As one can see, the results are quite varied for smaller assays, and the best performing algorithm is largely dataset dependent. Much of this variation is due to the threefold CV procedure of Mayr et al. that is quite susceptible to large variations because of the small dataset size.

However, as the training size increases, the deep learning and SVM algorithms dominate. Interestingly, among all datasets with greater than 1000 compounds in the test set, SVM performance is better than FNN performance on 62.5% of assays, which is counter to the usual wisdom that deep learning approaches beat SVMs in large assays. Notably, GC, LSTM, and Weave, show the best performance on only a small number of large assays, casting doubt on their utility over a standard FNN or SVM. With all of these observations, it should be noted that the results could be due to suboptimal hyperparameter optimization—and perhaps some of these models can achieve state-of-the-art performance in the hands of expert users. However, hyperparameter optimisation can take a considerable amount of time and computing resources.

Additionally, the correlation between the mean AUC–ROC performances for all models is shown in Fig. 4 for the 177 assays with more than 1000 test set samples on average over the threefold. As can be seen, most of the deep learning models perform quite similarly, with NB, KNN, and Weave seeming to show the worst performance. The figure showing how well the models correlate on assays of all sizes is also included in the supporting information. Unfortunately, it is tough to make inferences regarding the relative performance in small sizes due to the inherent noise of the datasets and threefold CV procedure.

Taking all of the above into consideration, it appears that the FNN and SVM models are the best performing models, especially in the case of large datasets. In small datasets, NB, KNN, and RF can often still perform competitively. It is

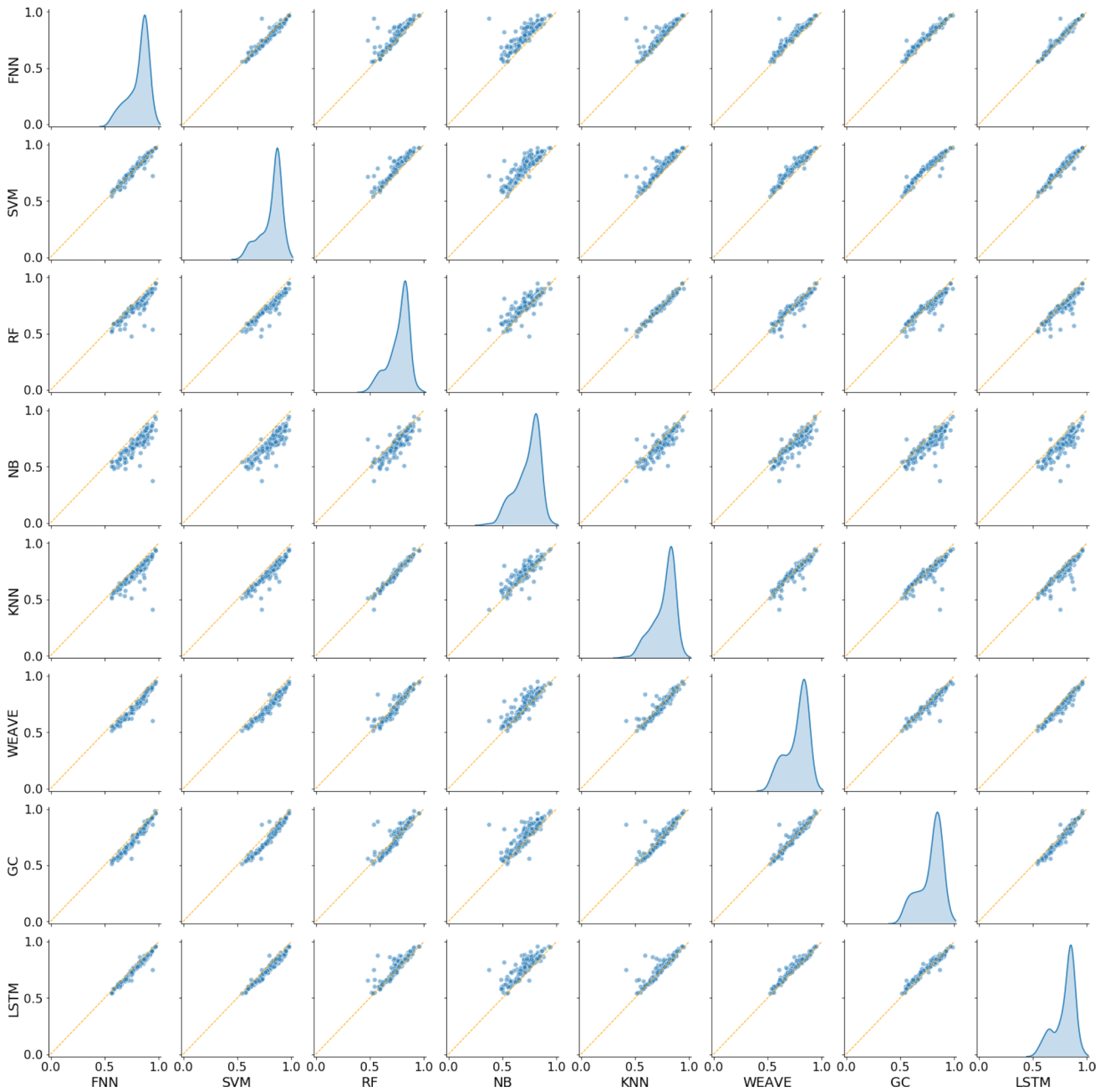

Fig. 4 The correlation in mean AUC–ROC performance for all models on assays with more than 1000 samples in the test set. The orange line indicates the identity line of slope 1, while the dot size indicates the size of the test set. The density plots on the diagonal represent the distribution of mean AUC–ROC scores for the given classifier

also unclear how well the frequently used gradient-boosted decision tree algorithm would compare, since it was not included in the study. The Mayr et al. data contains quite a lot of information and we provide it and our code online for all who wish to further analyze it.

What performance metric do we need?

Having reevaluated different machine learning methods using the metric suggested by Mayr et al. —ROC–AUC— we now turn to consider the more general question of what performance metric is most closely correlated to practical success in drug discovery.

Table 4 An example confusion matrix

		Predicted class	
		+	−
Actual Class	+	True positive (TP)	False negative (FN)
	−	False positive (FP)	True negative (TN)

In the drug discovery literature, the positive class (+) represents the actives, while the negative class (−) represents the inactives. Note that there are mixed conventions in the literature regarding the correct axis of the predicted and actual classes

What does ROC–AUC measure?

Typical machine learning performance measures such as accuracy, true positive rate (TPR), false positive rate (FPR), precision, specificity, are combinations of the entries of the confusion matrix, shown in Table 4.

In the virtual screening literature, researchers frequently use the area under the curve (AUC) of the receiver operating characteristic curve (ROC), which plots TPR vs FPR, defined below,

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN}$$

The AUC–ROC is not based on a single threshold, and instead gives an indication of classifier performance over a range of varying classification thresholds. In this way, the AUC–ROC captures how well a classifier discriminates between the classes of interest. Conveniently, the AUC–ROC can also be interpreted as the probability that a randomly chosen member of the positive class will be correctly ranked before a randomly chosen member of the negative class. Therefore, an AUC–ROC of 1.0 indicates perfect discrimination between classes, and an AUC–ROC of 0.5 indicates random guessing.

While the AUC–ROC is more robust than metrics such as accuracy in cases of class imbalance, it is still not without criticism. These critiques, popularized by Hand and coworkers [25], are perhaps best understood if one interprets the ROC–AUC as the expected TPR averaged over all classification thresholds (false positive rates). Therefore, if two ROC curves cross, the AUC of one curve may be higher even if it performs much worse (has a lower TPR) over the region containing the classification thresholds of interest. Additionally, Hand raises concerns about the "incoherence" of AUC–ROC, since the measure ignores relative cost concerns of each threshold when simply taking an expected value over all such decision thresholds (FPR from zero to one).

The critique of AUC–ROC most widely seen in drug discovery is that it does not account for the "early behavior" of a classifier. Since the purpose of virtual screening procedures is often to rank the compounds by likely activity and avoid experimentally screening an intractable number of compounds, the classifier is only useful if active compounds are ranked at the top of the list and prioritized for actual screening. Unfortunately, the AUC–ROC does not take into account this early performance and only measures average discrimination performance.

As a result of these shortcomings, many alternative methods including BedROC and RIE have been proposed, as described in [4]. However, these methods are sensitive to a tunable parameter and are not as interpretable as a metric as AUC–ROC. In drug discovery, the enrichment factor is often used to quantify this early behavior, which describes how many of the total actives are found in the top X% of ranked compounds. Unfortunately, this metric can be quite noisy and is sensitive to both the chosen percentage and specific ordering of compounds at the top of the list. Alternatives such as ROC enrichment, which instead uses the fraction of inactives and is related to the AUC–ROC, have also been proposed [4].

Instead of focusing on specific drug discovery metrics, we propose that the widely used area under the precision–recall curve (AUC–PRC) may serve as an important complement to the AUC–ROC in chemical applications. Precision–recall curves plot the precision or positive-predicted-value (PPV = $\frac{TP}{TP+FP}$) on the vertical axis and recall (same as TPR) on the horizontal axis. To illustrate why AUC–PRC may be more appropriate than AUC–ROC, we describe below a series of numerical simulations. Our simulations build on the results of Saito and Rehmsmeier [26].

Precision–recall should be used in conjunction with AUC–ROC

We first consider a theoretical classifier of positive and negative examples, shown in Fig. 5a. This theoretical classifier shows decent discrimination between the two classes, as shown by the separation of the two normal distributions.

Because this hypothetical classifier assigns scores to each class based on well known distributions, we can computationally take samples from each distribution. In our example, we take $N_+ = 100$ samples from the positive class and $N_- = 10,000$ samples from the negative class in order to mimic a 1% hit rate of actives that might be observed in a virtual screen. After repeating this experiment ten times, we then plot ROC curves, PRC curves, and percent enrichment factor curves for each round of the simulation, as shown in Fig. 5b–d.

The first thing to notice in these plots is the large discrepancy between the average AUC–ROC and AUC–PRC scores. This difference results from the large number of false positives which cause the precision–recall scores to be quite

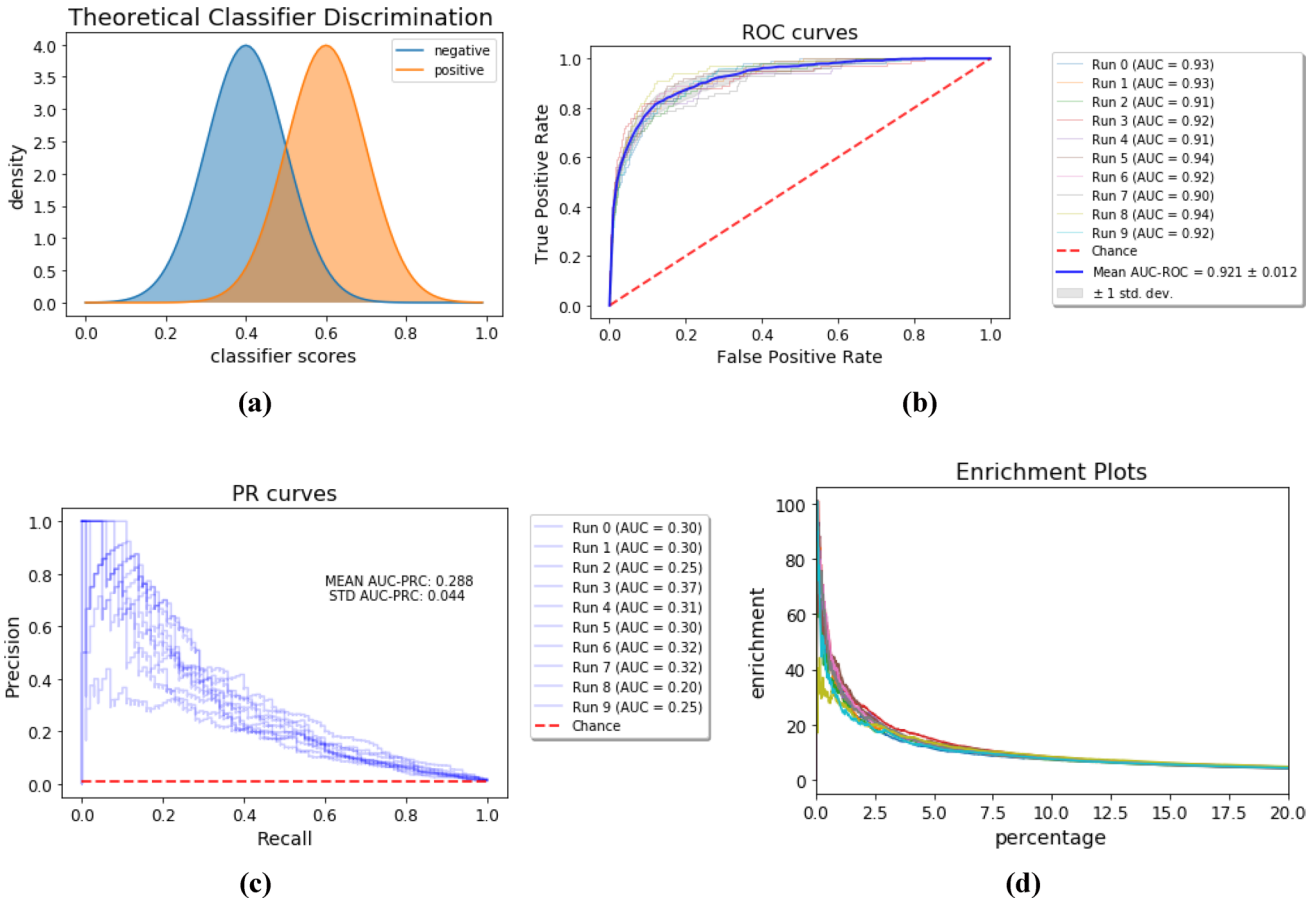

Fig. 5 **a** Distributions of classification scores for a theoretical classifier of positive and negative examples. The negative scores follow a $\mathcal{N}(\mu = 0.4, \sigma = 0.1)$ distribution, while the positive scores follow a $\mathcal{N}(\mu = 0.6, \sigma = 0.1)$. Adapted from the simulations in [26]. **b** ROC, **c** PRC, and **d** enrichment factor curves for predictions from the theoretical classifier shown in (**a**). The curves result from ten runs of a simulation with large class imbalance (∼ 1% actives)

low. Importantly, one may wrongly conclude that the performance of the classifier is almost perfect by merely observing the ROC plot, while observing the PRC plot indicates poor precision.

We next consider an alternate theoretical classifier with improved early performance in Fig. 6a. In this case, the negative samples are drawn from the same normal distribution as in the aforementioned simulation, while the positive samples are drawn from a Beta distribution to bias the results towards improved early retrieval of actives.

The same simulation procedure is repeated with the same class imbalance, and the results of the simulation are shown in Fig. 6b–d. Notably, the average AUC–ROC is almost the same as in the previous simulation but the average AUC–PRC and enrichment factors show a marked increase. Therefore, we observe that the precision–recall curve better accounts for the desired early performance behavior in drug-discovery applications. To show that this is not merely due to intricacies of our simulation setup, we replicate the simulations of Saito and Rehmsmeier, which show the same effect, in the supporting information.

We note that the "early-part" of the ROC curve (TPR values at low FPR values) also indicates the early performance. However, in benchmarks or cross-validation procedures, one does not often observe the complete ROC curve, and instead just observes the AUC–ROC values themselves.

As an example of the utility of the AUC–PRC score, we plot the AUC–ROC and AUC–PRC performances of FNN on all 1310 assays. As one can see in Fig. 7, the two metrics are not necessarily well correlated, and thus indicate large class imbalances. Additionally, observing that the AUC–PRC was often large when the AUC–ROC indicated mediocre performance alerted us to the fact that many of the assays show the opposite class skew we would expect from virtual screening assays. Importantly, a majority of the labeled compounds were active in a large number of assays (in 165 of the 1310 assays, there is at least one test fold consisting of 90% or greater

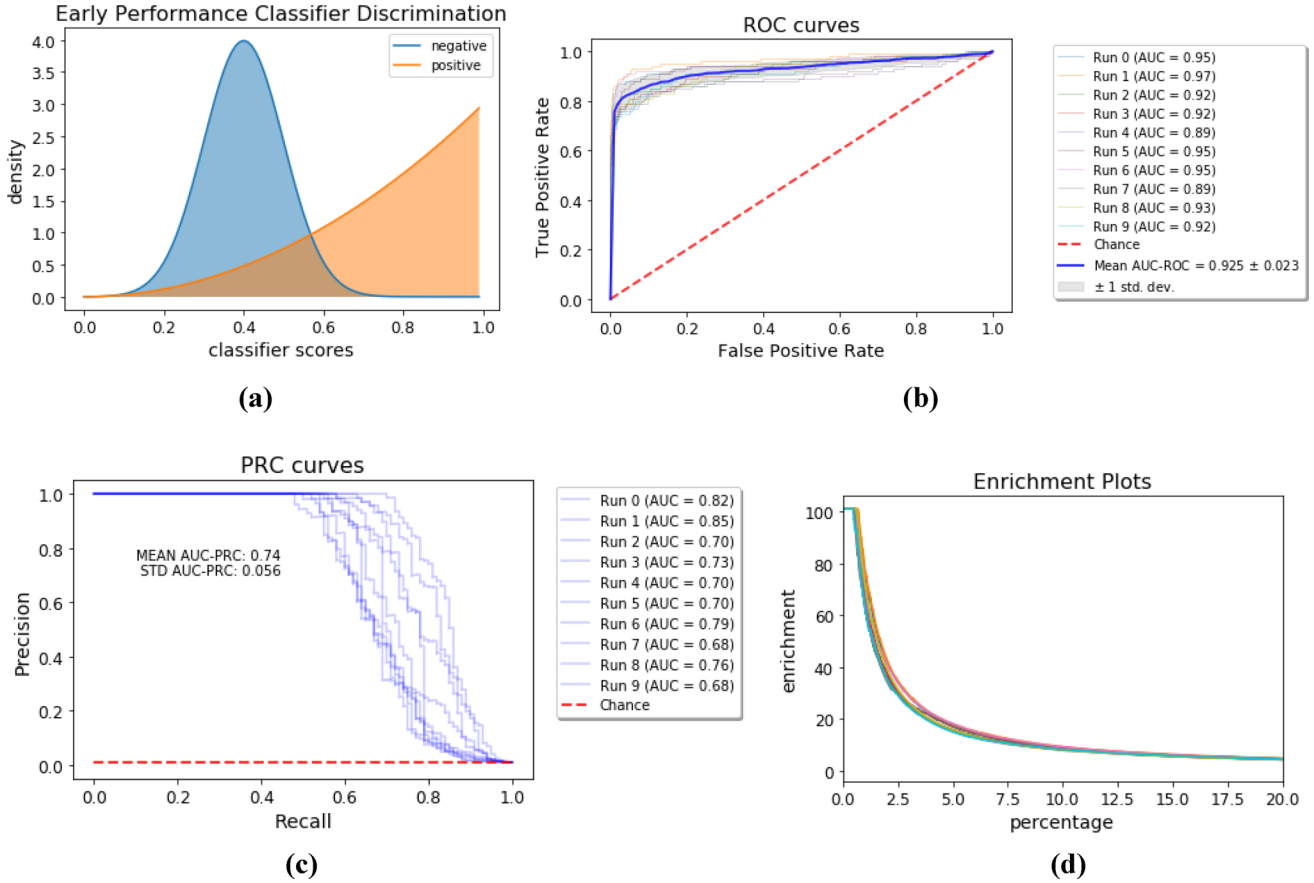

Fig. 6 **a** Distributions of classification scores for a theoretical classifier of positive and negative examples with improved early performance. The negative scores follow a $\mathcal{N}(\mu = 0.4,\ \sigma = 0.1)$ distribution as before in Fig. 5a, while the positive scores follow a $B(a = 3, b = 1)$ distribution. **b** ROC, **c** PRC, and **d** enrichment factor curves for predictions from the theoretical classifier shown in (a). The curves result from ten runs of a simulation with large class imbalance ($\sim 1\%$ actives)

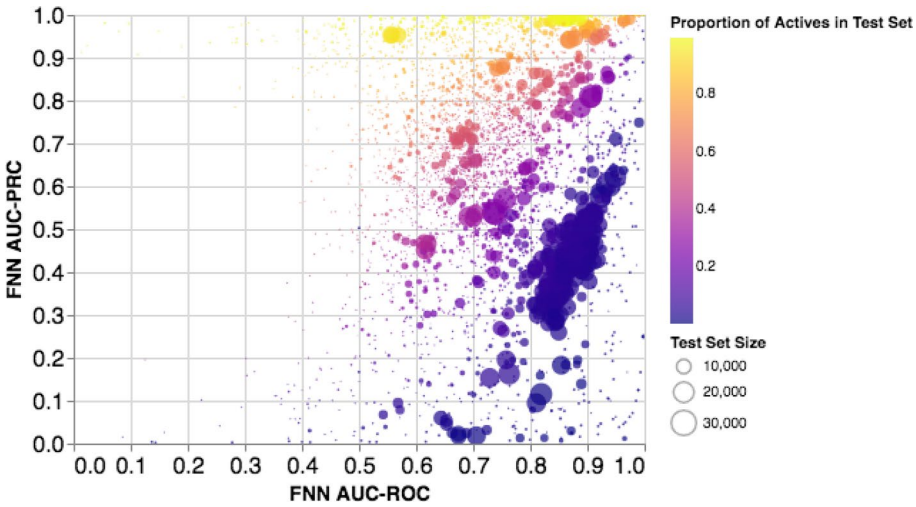

Fig. 7 The relationship between AUC–ROC scores and AUC–PRC scores for all test folds for the FNN model. Note the limited range of AUC–ROC scores in comparison to the AUC–PRC scores, which vary considerably. We can only compare ROC and PRC for FNN because PRC values were not reported for the other methods

actives), even though virtual screening applications usually involve situations where the inactives far outnumber the active hits (in nearly all collected and literature published datasets, molecular diversity is generally greater and not focused on an active series or related molecules).

Understanding what we want to measure

The above discussion details how AUC–PRC may be useful for evaluating the performance of computational methods in drug discovery. However, this is not to say that precision–recall is always the correct metric. In general, the correct metric is largely dependent on the problem of interest and the associated costs of false positives and false negatives.

As with AUC–ROC, there are known problems with use of the precision–recall curve. For example, Boyd and coworkers have shown that there are "unachievable regions" of the PRC space [27]. Flach and Kull have further suggested the alternative precision–recall–gain curve for fixing incoherence issues and improving model selection [28]. Additionally, we note that the AUC–PRC of a random classifier ($\frac{N_+}{N_+ + N_-}$) is dependent on the number of positive and negative samples in a dataset, whereas the AUC–ROC of a random classifier is always 0.5.

Appreciating these concerns, the metric of interest must be chosen after carefully considering the problem of interest and the quality of the associated data. One clear difference between the ROC and PRC curves is the consideration of true negatives. While true negatives greatly impact the AUC–ROC because they factor into the calculation of FPR, the AUC–PRC metric does not consider the correct classification of negative/inactive samples. Therefore, if the inactives are uninformative, use of AUC–ROC may be dangerous. In the particular domain of drug discovery, inactives may be uninformative because they are mere "assumed inactives," because they are deemed inactive by an arbitrarily chosen activity cutoff value, or because they show inactivity for any number of unknown unknowns (e.g. the molecules are simply insoluable). Thus, selecting models based on superior AUC–ROC performance may result in models that are best at classifying negative samples, even though that is not the behavior of interest.

How should models be trained and tested?

Another major issue in the chemical machine learning literature that Mayr and coworkers hoped to address was the way models are trained and tested. They combined two innovative methodologies – nested cross validation and scaffold splitting. We will summarise those methodologies below, and then discuss the tradeoffs.

Cross-validation and scaffold splitting

Machine learning models often contain many hyperparameters. Cross-validation (CV) is a strategy that enables the user to tune those hyperparameters without overfitting. The nested CV protocol consists of an outer and inner CV loop. The entire procedure is perhaps best understand as a simple k-fold CV procedure, in which the holdout test set is one of n distinct folds. The $n \times k$ nested CV setup thus consists of n different simple k-fold CV procedures for model selection, followed by model evaluation on the n distinct testing folds. Accordingly, the average of the performance on the n testing folds provides an almost completely unbiased estimate of the true generalization performance [13, 29].

However, cross validation alone is insufficient to estimate the true performance of models. Compounds in chemical datasets are often centered around easily synthesized scaffolds, which are then modified by adding functional groups. Therefore, certain machine learning algorithms may just memorize properties of certain scaffolds and fail to generalize to new chemicals. Thus, if similar compounds are contained in both the training and test sets, we expect that estimates of machine learning performance would overestimate the true generalization performance on new compounds.

To counteract this problem, it is now popular to perform "scaffold splitting", where compounds are split into subsets based on their two-dimensional structural framework. An implementation is included in the DeepChem package [30], and the results of random splitting have been explored both in the MoleculeNet benchmark paper, wherein Wu and coworkers reported larger differences between training and testing performance with scaffold splitting than with random splitting, as is expected.

However, we note that active compounds of different scaffolds may interact with a given target through a different mode of action. Therefore, expecting a model to generalize by learning from other scaffolds may be unrealistic.

Table 5 The results of an FNN deep learning model on the ChEMBL 1243971 assay. The AUC–ROC scores for the three disjoint test folds are reported

	Fold 1	Fold 2	Fold 3
AUC–ROC	0.69	0.00	0.56
Test sen Number Actives/inactives	18/18	2/1	3/3

Averages and outliers

We return now to examining the specific model evaluation and comparison approaches found in Mayr et al. Generally, their approach involves performing the aforementioned 3×2 nested cluster-cross-validation model evaluation procedure on all 1310 assays. Importantly, due to the setup of their clustering approach, the number of compounds in each fold is not the same. Furthermore, the ratio of active to inactive compounds in each fold may be quite different. For example, consider the FNN results shown in Table 5 on a particularly troublesome assay CHEMBL1243971, which measures the inhibition of the PI4-kinase beta subunit. This assay is one of the smallest in the entire dataset, and includes folds that are heavily imbalanced in terms of size. In order to compare the performance of FNN to other models on this assay, the mean and standard deviation AUC–ROC scores over all threefold were calculated by Mayr et al. However, this averaging completely discards the inherent uncertainty of each independent test fold, which can be useful information.

Taking fold 2 as an example, we can take the approach of [31] and recognize that the AUC–ROC, which is again the probability that a randomly chosen positive sample is correctly ranked higher than a randomly chosen negative sample, is equivalent to the value of the Wilcoxon-Mann-Whitney statistic. Doing this calculation, we find that a classifier can achieve three possible values of AUC–ROC with two active and one inactive compound in the test set, AUC–ROC = 0.0, AUC–ROC = 0.5, and AUC–ROC = 1.0 . Therefore, a completely random classifier, would achieve a mean AUC of 0.5 on fold 2 with a standard deviation of 0.41. Thus, most confidence intervals of interest would include the entire possible range of AUC values, [0, 1], and any result on this fold is essentially meaningless. Even the larger, more balanced, fold 1 has a relatively large 95% confidence interval of AUC–ROC $\in [0.49, 0.84]$ using the approximation in [3] and thus cannot be rejected as a random classifier (AUC–ROC = 0.5) at the $\alpha = 0.05$ significance threshold. Importantly, since averaging treats all folds the same, the result on fold 1 is treated as of equal importance to the meaningless result on fold 2. Furthermore, one outlier fold performance can significantly affect the average AUC on a given assay. Thus, the use of unequal fold sizes coupled with averaging AUC scores renders interpretation challenging.

Cross-validation underestimates error

Leaving the concern of variable amounts of data for each fold aside, a fundamental question is whether cross-validation provides an accurate measure of error. Varoquaux previously performed extensive simulations to show that the standard error across cross-validation folds considerably underestimates the actual error [32]. To understand the extent of this error, we adapted his simulation procedures focusing on prediction accuracy to measure the systematic errors in AUC–ROC estimation from cross-validation. All of our code for these simulations has been made available in the supplementary information.

We begin by constructing an artificial high-dimensional dataset of two relatively well separated Gaussian distributions. The separation of the distributions is adjusted such that a Linear SVM classifier will achieve an AUC–ROC = 0.75 on the data. A training set of size N_{train} is then drawn from the artificial two-class dataset. The Linear SVM is trained on these N_{train} samples then deployed on a test set of size 10,000 samples, which gives an estimate of the true generalization performance. This true generalization performance is compared to the mean performance of a threefold CV procedure on the original N_{train} training samples. Thus, we can compare the classifier's true generalization AUC–ROC performance to the estimate of that performance from cross-validation.

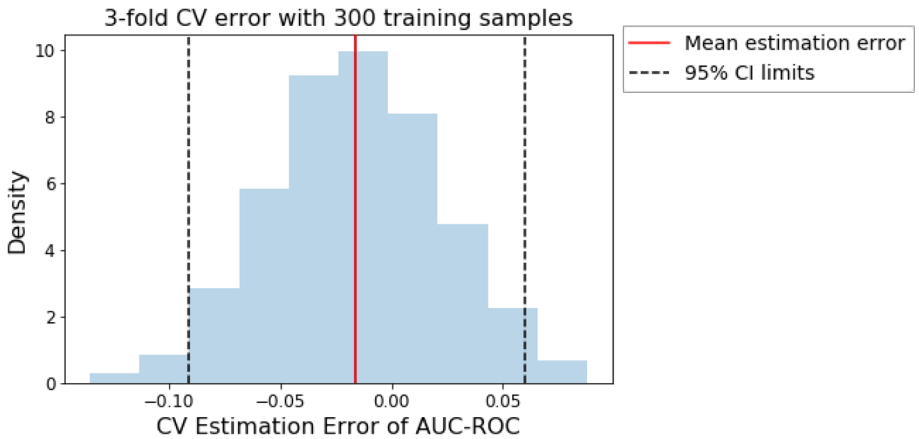

Fig. 8 The distribution of the errors when estimating true generalization performance from threefold CV for $N_{\text{train}} = 300$ training samples. The dotted black lines indicate the 2.5 and 97.5 percentiles, thus denoting the boundaries of a 95% confidence interval

Figure 8 shows the results from 1000 runs of the simulation using $N_{\text{train}} = 300$ training samples. As can be seen, the CV estimates of performance are still frequently off by over 0.05 in AUC–ROC space. Moreover, the distribution of errors is asymmetric, as is to be expected for AUC–ROC. It should be noted that these large errors result even in the case of an artificial dataset of well-behaved distributions and equal sized folds with small class imbalance.

In addition to measuring CV errors in estimation of generalization performance, we can use these simulation results to measure how well the standard error of the mean (SEM) represents the variability of the CV procedure. To measure this, we construct 95% confidence intervals centered around the mean 3-fold CV performance. Theoretically, 95% of the confidence intervals constructed in this fashion should contain the true generalization performance if the cross-validation procedure is unbiased. However, we find that the confidence interval coverage is only 79.7% for $N_{\text{train}} = 300$, even when using the extremely generous bounds to the t-distribution with 2 degrees of freedom.

The coverage is similarly poor for other sizes of N_{train}, and would be much worse if one naively used the $\pm 2 \times (SEM)$ rule to construct confidence intervals. These results unfortunately indicate that the confidence intervals on cross validation means are often too optimistic. This bias results from the correlation of training data across folds, thus violating independence assumptions. These simulation results cast doubt on the ability of cross validation procedures with small sample sizes to accurately reflect the generalization performance of a classifier with appropriate uncertainty.

As Varoquaux notes, these results are particularly galling since cross-validation seems to be our current best tool to estimate model performance. Like Varoquaux, we have no specific suggestions for alternatives, but rather hope that others come to understand this uncertainty.

When benchmarks such as MoleculeNet [21] or the work examined herein report results as confidence intervals from cross-validation procedures, these estimates are likely underestimates of the true variability. Furthermore, simply averaging results across cross-validation folds may mask the uncertainty in the results of each individual fold. There are indeed multiple types of uncertainty such as that arising from model training, dataset splitting, the sometimes arbitrary delineation of actives/inactives, etc. And unfortunately, not all of these uncertainties are likely incorporated into confidence intervals. Only once we respect these limitations, will the gap between performance estimates in the literature and actual results in the lab/clinic begin to narrow.

Conclusion

We build on the recent large-scale benchmarking study by Mayr and coworkers and reanalysed the reported performance data of different machine learning models, arriving at a different conclusion to Mayr and coworkers. We show that support vector machines achieve competitive performance compared to feed-forward deep neural networks. Moreover, we show, via numerical simulations, that the area under the precision–recall curve can be more informative than the area under the receiver operating characterise curve in terms of assessing the performance of machine learning models in contexts relevant to drug discovery. We also highlight challenges in interpreting scaffold splitting cross validation results.

All of these results show a clear need for further research into proper validation procedures for chemoinformatics. We have demonstrated that the current approach of examining average performance across diverse assays ignores great uncertainty in our evaluation procedures. Furthermore, the current approach does not yield special insight into which models are best suited for which classes of problems. Each model has its own inductive biases, and we propose more research into understanding when each model will work/fail on chemical data. This future work will likely require focus on high-quality, representative data, rather than large amounts of heterogeneous data.

Acknowledgements AAL acknowledges the support of the Winton Programme for the Physics of Sustainability. The authors would like to thank the anonymous referees for the their insightful comments and sugguestions.

References

1. Walters WP (2013) J Chem Inf Model 53:1529. https://doi.org/10.1021/ci400197w
2. Landrum GA, Stie N (2012) Future Med Chem 4:1885
3. Nicholls A (2014) J Comput-Aided Mol Des 28:887
4. Nicholls A (2008) J Comput-Aided Mol Des 22:239
5. Nicholls A (2016) J Comput-Aided Mol Des 30:103
6. Wallach I, Heifets A (2018) J Chem Inf Model 58:916
7. Sieg J, Flachsenberg F, Rarey M (2019) J Chem Inf Model 59:947

8. Santafe G, Inza I, Lozano JA (2015) Artif Intell Rev 44:467
9. Derrac J, García S, Molina D, Herrera F (2011) Swarm Evolut Comput 1:3
10. Dietterich TG (1998) Neural Comput 10:1895
11. Demšar J (2006) J Mach Learn Res 7:1
12. Japkowicz N, Shah M (2011) Evaluating learning algorithms: a classification perspective. Cambridge University Press, Cambridge
13. Raschka S (2018) arXiv preprint arXiv:1811.12808
14. Recht B, Roelofs R, Schmidt L, Shankar V (2018) CoRR arXiv :abs/1806.00451
15. Border R, Johnson EC, Evans LM, Smolen A, Berley N, Sullivan PF, Keller MC (2019) Am J Psychiatry 176(5):376–387
16. Rogers D, Hahn M (2010) J Chem Inf Model 50:742
17. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) In: 2009 IEEE conference on computer vision and pattern recognition (IEEE) pp 248–255
18. Niven T, Kao H (2019) CoRR arXiv:abs/1907.07355
19. Mayr A, Klambauer G, Unterthiner T, Steijaert M, Wegner JK, Ceulemans H, Clevert D-A, Hochreiter S (2018) Chem Sci 9:5441
20. Goh GB, Hodas NO, Vishnu A (2017) J Comput Chem 38:1291
21. Wu Z, Ramsundar B, Feinberg EN, Gomes J, Geniesse C, Pappu AS, Leswing K, Pande V (2018) Chem Sci 9:513
22. Bento AP, Gaulton A, Hersey A, Bellis LJ, Chambers J, Davies M, Krüger FA, Light Y, Mak L, McGlinchey S et al (2014) Nucleic Acids Res 42:D1083
23. Landrum G et al (2006) Rdkit: open-source cheminformatics
24. Ruscio J (2008) Psychol Methods 13:19
25. Hand DJ (2009) Mach Learn 77:103
26. Saito T, Rehmsmeier M (2015) PLoS ONE 10:e0118432
27. Boyd K, Costa VS, Davis J, Page D (2012) CoRR arXiv :abs/1206.4667
28. Flach P, Kull M (2015) Precision-recall-gain curves: PR analysis done right. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems, vol 28. Curran Associates, Inc., New York, pp 838–846
29. Varma S, Simon R (2006) BMC Bioinform 7:91
30. Democratizing deep-learning for drug discovery, quantum chemistry, materials science and biology (2016) https://github.com/deepchem/deepchem
31. Hanley JA, McNeil BJ (1982) Radiology 143:29
32. Varoquaux G (2018) Neuroimage 180:68